THAT WHICH IS CLAIMED:

1. A bit block for an adder, the bit block comprising:

a first bit stage that generates a first bit associated propagation characteristic (bapc) that is independent of a carry input to the bit block from another bit block of the adder.

2. The bit block of Claim 1 further comprising:

a second bit stage that, based on the first bapc, generates a second bapc that is independent of the carry input to the bit block.

3. The bit block of Claim 2 wherein:

the first bit stage generates the first bapc based on first and second operand bits input to the first bit stage and a third bapc that is generated by a less significant bit stage of the bit block and is independent of the carry input to the bit block; and

the second bit stage generates the second bapc based on first and second operands input to the second bit stage.

4. The bit block of Claim 3 wherein the first and second bit stages each further generates a respective sum bit based on its input first and second operand bits and a respective first and second bit stage carry input.

5. The bit block of Claim 4 wherein the second bit carry input to the second bit stage is generated by the first bit stage, the first bit stage selecting either the carry input to the bit block or a calculated carry output as the second bit carry input based on the third bapc.

6. The bit block of Claim 5 further comprising:

a least significant bit stage that generates an initial bapc that is independent of the carry input to the bit block based on first and second operands input to the

least significant bit stage.

7. The bit block of Claim 6 wherein the least significant bit stage generates the initial bapc as the exclusive nor of the first and second operands input to the least significant bit stage.

8. The bit block of Claim 7 wherein the least significant bit stage further generates a sum bit and a carry output based on the carry input to the bit block and the first and second operands input to the least significant bit stage.

9. The bit block of Claim 8 wherein the adder is a carry-skip adder and wherein the bit block comprises one of a plurality of bit blocks of the carry-skip adder.

10. The bit block of Claim 8 wherein the bit block further comprises a most significant bit stage configured to generate a last bapc that is independent of the carry input to the bit block, the last bapc being provided as a skip select signal output from the bit block in the carry-skip adder.

11. The bit block of Claim 8 wherein the adder is a hybrid carry-look-ahead adder and wherein the bit block is included in an adder stage of the adder coupled to a carry tree of the adder.

12. The bit block of Claim 11 wherein the carry tree of the adder comprises a Lynch-Swartzlander type carry tree and wherein the adder stage is substituted for the carry select circuit of a Lynch-Swartzlander type hybrid carry-look-ahead adder.

13. The bit block of Claim 11 wherein the carry tree of the adder comprises a Kantabutra type carry tree and wherein the adder stage is substituted

for the carry select circuit of a Kantabutra type hybrid carry-look-ahead adder.

14. The bit block of Claim 5 wherein the first bit stage is further configured to calculate the calculated carry output responsive to input of the first and second operand bits to the first bit stage without waiting for input of the carry input to the bit block.

15. The bit block of Claim 5 wherein the bit block comprises an eight bit block including 8 bit stages.

16. The bit block of Claim 15 wherein the adder comprises a 32-bit adder.

17. The bit block of Claim 5 wherein the adder comprises a 32-bit adder.

18. The bit block of Claim 5 wherein the first and second bit stage each further comprise a three input multiplexer.

19. The bit block of Claim 18 wherein the three input multiplexer of a respective bit stage includes a first input coupled to one of the first and second operands of the respective bit stage, a second input coupled to the carry input to the bit block, a third input coupled to a calculated carry output of a preceding bit stage of the bit block, a first select input coupled to an exclusive nor of the first and second operands of the respective bit stage and a second select input coupled to the bapc generated by the respective bit stage.

20. The bit block of Claim 19 wherein the three input multiplexer provides the output logic OUT = S0A + $\overline{S0}\,\overline{S1}$ B + $\overline{S0}\,S1$ C + $S0\,\overline{S1}$ B wherein A, B and C are the inputs and S0 and S1 are select inputs.

21. The bit block of Claim 11 wherein the adder stage has a worst case delay no greater than a worst case delay of the carry tree of the adder.

22. The bit block of Claim 11 wherein the adder comprises a 56-bit operand adder.

23. The bit block of Claim 13 wherein the adder stage comprises a 16-bit adder and the bit block comprises either a four bit block or a six bit block.

24. The bit block of Claim 13 wherein the adder stage comprises a 15-bit adder and the bit block comprises either a four bit block or a five bit block.

25. The bit block of Claim 24 wherein the bit block further comprises an external propagation characteristic input that indicates whether a carry output for a bit stage of the bit block is to be generated from within the adder including the bit block or is dependent on a carry input to the adder.

26. An adder comprising:

a first bit block; and

a second bit block that receives a block carry input from the first bit block; and

wherein the second bit block comprises a first bit stage that generates a first bit associated propagation characteristic (bapc) that is independent of the block carry input from the first bit block.

27. The adder of Claim 26 wherein the second bit block further comprises a second bit stage that, based on the first bapc, generates a second bapc that is independent of the block carry input from the first block.

28. The adder of Claim 27 wherein:

the first bit stage generates the first bapc based on first and second operand bits input to the first bit stage and a third bapc that is generated by a less significant bit stage of the bit block and is independent of the block carry input from the first bit block; and

the second bit stage generates the second bapc based on first and second operands input to the second bit stage.

29. The adder of Claim 28 wherein the first and second bit stages each further generate a sum bit based on their input first and second operand bits and a respective first and second bit stage carry input.

30. The adder of Claim 29 wherein the second bit carry input to the second bit stage is generated by the first bit stage, the first bit stage selecting either the block carry input from the first bit block or a calculated carry output as the second bit carry input based on the third bapc.

31. The adder of Claim 30, the second bit block further comprising:

a least significant bit stage that generates an initial bapc that is independent of the block carry input from the first bit block based on first and second operands input to the least significant bit stage.

32. The adder of Claim 31 wherein the least significant bit stage generates the initial bapc as the exclusive nor of the first and second operands input to the least significant bit stage.

33. The adder of Claim 31 wherein the adder is a carry-skip adder.

34. The adder of Claim 33 wherein the second bit block further comprises a most significant bit stage configured to generate a last bapc that is

independent of the block carry input from the first bit block, the last bapc being provided as a skip select signal output from the second bit block in the carry-skip adder.

35. The adder of Claim 31 wherein the adder is included in a hybrid carry-look-ahead adder and wherein the adder is coupled to a carry tree of the hybrid carry-look-ahead adder.

36. The adder of Claim 35 wherein the carry tree of the hybrid carry-look-ahead adder comprises a Lynch-Swartzlander type carry tree and wherein the adder is substituted for the carry select circuit of a Lynch-Swartzlander type hybrid carry-look-ahead adder.

37. The adder of Claim 35 wherein the carry tree of the hybrid carry-look-ahead adder comprises a Kantabutra type carry tree and wherein the adder is substituted for the carry select circuit of a Kantabutra type hybrid carry-look-ahead adder.

38. The adder of Claim 30 wherein the first bit stage is further configured to calculate the calculated carry output responsive to input of the first and second operand bits to the first bit stage without waiting for input of the block carry input from the first bit block.

39. The adder of Claim 30 wherein the second bit block comprises an eight bit block including 8 bit stages.

40. The adder of Claim 39 wherein the adder comprises a 32-bit adder.

41. The adder of Claim 35 wherein the adder has a worst case delay no greater than a worst case delay of the carry tree of the hybrid carry-look-ahead

adder.

42. The bit block of Claim 35 wherein the adder comprises a 56-bit operand adder.

43. The bit block of Claim 37 wherein the adder stage comprises a 16-bit adder and the bit block comprises at least one of a four bit block and a six bit block.

44. The bit block of Claim 37 wherein the adder stage comprises a 15-bit adder and the bit block comprises either a four bit block or a five bit block.

45. A method for adding operands in a first bit block of an adder which receives a block carry input from a second bit block of the adder, the method comprising:

providing a least significant bit stage and a plurality of other bit stages in the first bit block;

generating a first bit associated propagation characteristic (bapc) from a first of the other bit stages based on bits of the operands input to the first of the other bit stages and a bapc generated by the least significant bit stage, the first bapc being independent of the block carry input;

generating a second bapc from a second of the other bit stages based on the first bapc and bits of the operands input to the second of the other bit stages, the second bapc being independent of the block carry input; and

adding the operands based on the first and second bapc and bits of the operands input to the first bit block.

46. The method of Claim 45 further comprising generating a first carry output from the first of the other bit stages that provides a carry input to the second of the other bit stages and wherein the step of adding the operands includes:

generating a sum output from the second of the bit stages based on the bits of the operands input to the second of the other bit stages, the first bapc and the carry input to the second of the other bit stages; and

generating a second carry output from the second of the other bit stages that provides a carry input to a third of the other bit stages, the second carry output being selected as either the carry input to the second of the other bit stages or calculated based on at least one of the bits of the operands input to the second bit stage based on the first bapc.

47. The method of Claim 46 wherein the step of generating a second carry output further comprises calculating the second carry output responsive to input of the bits of the operands input to the second of the other bit stages without waiting for input of the carry input to the second of the other bit stages.

48. The method of Claim 47 further comprising generating the bapc generated by the least significant bit stage based on bits of the operands input to the least significant bit stage independent of the block carry input.

49. The method of Claim 47 wherein the step of generating the bapc generated by the least significant bit stage as an exclusive nor of the bits of the operands input to the least significant bit stage.

50. The method of Claim 48 wherein the first and second bit block are included in a carry-skip adder, the method further comprising generating a last bapc that is independent of the carry input to the bit block from a most significant bit stage of the other bit stages, the last bapc being provided as a skip select signal output from the first bit block in the carry-skip adder.

51. A method for selecting block sizes for n bit blocks of an N bit carry-skip adder, the method comprising:

determining a skip time (s) and a ripple time (r) for the carry-skip adder;

identifying a set of right triangles (Δ_ρ) having a base defining an axis representing a block number of ones of the bit blocks, wherein block number 0 corresponds to a most significant one of the bit blocks and increasing bit block numbers correspond to decreasingly significant ones of the bit blocks, the right triangles further having a vertical left side paralleling a vertical axis representing a number of bits in the respective ones of the bit blocks and a right side having a slope of $-\sigma$;

selecting sets of bit sizes for all except block number 0 that lie substantially on or within respective ones of the set of right triangles; and

selecting one of the sets of bit sizes corresponding to a smallest one of the set of right triangles, wherein a cumulative total of bits represented by the selected set of bit sizes and an associated number of bits of block number 0 contains at least N bits.

52. The method of Claim 51 wherein the associated number of bits of block number 0 for ones of the sets of bit sizes lies within or slightly above the corresponding respective one of the set of right triangles.